

# Managed Services Plugin

In order to add 3rd party service and a new role definition you need to drop in the service plug-in in sipXconfig jar directory. <http://track.sipfoundry.org/browse/XX-2115>

Service plug-in is similar to phone plug-in: but instead of new phone you describe a new service (SipxService) and optionally a new bundle (SipxServiceBundle). You can have a single plugin adding multiple services (check service.beans.xml and bundle.beans.xml for example definitions).

## Defining new service

Example: **Openfire Service**

**Step 1.** Create project directory

Under sipXconfig/plugins directory: create a new project. For example: **openfire**

**Step 2.** Create package

Create a new package. For example: **org.sipfoundry.sipxconfig.openfire**

**Step 3.** Create implementation.

Inside the package:

Create the service java class. It must extend **org.sipfoundry.sipxconfig.service.SipxService** otherwise system will not recognize your class as a possible service.

```
public class SipxOpenfireService extends SipxService {
    public static final String BEAN_ID = "sipxOpenfireService";
}
```

**Step 4.** Create plugin manifest

Create the xml spring configuration file: **sipxplugins.beans.xml** located in the root of your source tree

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:util="http://www.springframework.org/schema/util"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans-2.0.xsd
    http://www.springframework.org/schema/util
    http://www.springframework.org/schema/util/spring-util-2.0.xsd">
  <bean id="imBundle" parent="abstractBundle">
    <constructor-arg value="im" />
  </bean>
  <bean id="sipxOpenfireService" class="org.sipfoundry.sipxconfig.openfire.SipxOpenfireService" scope="
prototype" parent="sipxService">
    <property name="processName" value="SipXopenfire" />
    <property name="bundles">
      <set>
        <ref bean="imBundle" />
      </set>
    </property>
  </bean>
</beans>
```

Any other Spring beans that perform any kind of functionality should be added here.

**Step 5.** Create service start/stop script

Create a corresponding process.xml file that should be added on the sipX installation directory in process.d directory (near the other process.xml files): **sipxopenfire-process.xml**

```

<?xml version='1.0' encoding='iso-8859-1' standalone='yes'?>
<sipXecs-process xmlns='http://www.sipfoundry.org/sipX/schema/xml/sipXecs-process-01-00'>
  <name>SipXopenfire</name>
  <version>@VERSION@</version>
  <commands>
    <configtest>
      <execute>@SIPX_BINDIR@/sipxopenfire.sh</execute>
      <parameter>--configtest</parameter>
    </configtest>
    <start>
      <execute>@SIPX_BINDIR@/sipxopenfire.sh</execute>
      <parameter>--start</parameter>
    </start>
    <stop>
      <execute>@SIPX_BINDIR@/sipxopenfire.sh</execute>
      <parameter>--stop</parameter>
    </stop>
  </commands>
  <status>
    <pid>@SIPX_RUNDIR@/sipxopenfire.pid</pid>
    <log>@SIPX_LOGDIR@/sipxopenfire.log</log>
  </status>
  <resources>
  </resources>
</sipXecs-process>

```

Be sure that

`<property name="processName" value="SipXopenfire" />` in `sipxplugins.beans.xml` matches

`<name>SipXopenfire</name>` in your service start/stop script.

#### Step 6. Packaging

Update appropriate Makefile, and rpm spec file to build and install your plugin.